Automatic 3d free form shape matching using the graduated assignment algorithm.

Liu, Yonghuai

Published in:
Pattern Recognition

DOI:
10.1016/j.patcog.2005.01.008

Publication date:
2005

Citation for published version (APA):
Liu, Y. (2005). Automatic 3d free form shape matching using the graduated assignment algorithm. *Pattern Recognition*, *38*(10), 1615-1631. https://doi.org/10.1016/j.patcog.2005.01.008

Automatic 3D Free Form Shape Matching Using the Graduated Assignment Algorithm

Yonghuai Liu *
Department of Computer Science
University of Wales, Aberystwyth
Ceredigion SY23 3DB, Wales, UK
Email: yyl@aber.ac.uk

Abstract

3D free form shape matching is a fundamental problem in both the machine vision and pattern recognition literatures. However, the automatic approach to 3D free form shape matching still remains open. In this paper, we propose using k closest points in the second view for the automatic 3D free form shape matching. For the sake of computational efficiency, the optimised k-D tree is employed for the search of the k closest points. Since occlusion and appearance and disappearance of points almost always occur, slack variables have to be employed, explicitly modelling outliers in the process of matching. Then the relative quality of each possible point match is estimated using the graduated assignment algorithm, leading the camera motion parameters to be estimated by the quaternion method in the weighted least squares sense. The experimental results based on both synthetic data and real images without any pre-processing show the effectiveness and efficiency of the proposed algorithm for the automatic matching of overlapping 3D free form shapes with either sparse or dense points.

Keywords: 3D free form shape, Automatic matching, K closest points, Graduated assignment, Optimised k-D tree, Time complexity, Space complexity

1 Introduction

Recent technological development in electronics and optics enables the advent of laser scanning systems which capture geometrical and/or optical information in the form of range and/or intensity images of the object of interest in 3D space. Range images include the depth information of the object of interest from the laser scanning systems (Figures 4 and 6) and thus, greatly facilitate research on 3D free form shape matching. Since the laser scanning systems (range cameras) have limited field of view, a number of images have to be captured from different viewpoints so that a full coverage of the object surface can be obtained. All these images are depicted in local laser scanning system centred coordinate frame. For the construction of a full model of the

*Corresponding author. Tel: +44 1970 621688, Fax: +44 1970 628536

object, all these images have to be aligned in a single coordinate frame. This process is called registration, alignment, or matching (based on points).

3D free form shape matching has attracted much attention from both the machine vision and pattern recognition communities, as it is a fundamental problem for numerous applications of the latest laser scanning technologies in the areas like object recognition, motion estimation, scene understanding, and computer aided geometric design (CAGD). Free form shape matching has two goals: one is to determine correspondences between different data sets representing the same free form shape from different viewpoints, the other is to estimate the camera motion parameters bringing one data set into alignment with the other. Once one goal has been fulfilled, then the other is relatively easier. However, these two goals are in practice often interwoven, thus complicating free form shape matching.

1.1 Related work

In the last decade, a large number of techniques have been proposed to tackle the fundamental and challenging 3D free form shape matching problem based on techniques, such as scatter matrix [1], iterative closest point (ICP) [2, 3, 4], improved ICP algorithm [5, 6, 7, 8, 9, 10, 12, 11, 13, 14, 15, 16], interactive method [17], geometric histogram [18], and graduated assignment algorithm [19, 20, 21, 22] among many others. Among these methods, the ideas of the ICP algorithm and the graduated assignment algorithm are the most attractive and their brief analysis is thus given as follows.

1.2 Analysis of ICP

The ICP algorithm assumes that given the initial camera motion parameters rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$, for any point $\mathbf{p}$ in the first image, the closest point $\mathbf{p}'$ in the second image to the transformed point $\mathbf{Rp}+\mathbf{t}$ is its possible correspondent. This criterion is so practical and efficient that the ICP algorithm has become a *de facto* standard method for 3D free form shape matching. However, due to inaccurate initial camera motion parameters, occlusion and appearance and disappearance of points, the closest point criterion unavoidably introduces false matches in almost every iteration of matching. As a result, the key to successfully applying the closest point criterion for 3D free form shape matching lies in eliminating false matches. Unfortunately, research has shown [8, 16] that it is very difficult, if not impossible, to accurately evaluate the possible point matches established by the closest point criterion.

The false matches introduced by the ICP criterion are, to some degree, caused by the ignorance of a two-way constraint: if point $\mathbf{p}'$ is the correspondent of point $\mathbf{p}$, then point $\mathbf{p}$ should also be the correspondent of point $\mathbf{p}'$. However, in practice, once this two-way constraint is enforced, the matching algorithm often becomes much more complex not only in establishing possible point correspondences, but the optimisation of camera motion parameters as well, given the possible point correspondences between the free form shapes to be matched.

1.3 Analysis of the graduated assignment algorithm

In [20], a graduated assignment (GA) algorithm was proposed for 3D free form shape matching. Recently, it was extended for the alignment of a 3D model and a 2D projective image [23]. The

advantages of the GA algorithm lie in that:

1. It maximises entropy for the estimation of the probability of each possible point correspondence and the entropy maximization leads to the least biased estimate of probability distributions possible on the given information [24];

2. While it explicitly models outliers in the process of 3D free form shape matching, it adopts the Sinkhorn iterative alternate row and column normalization procedure [25] to gradually impose the two-way constraint, greatly simplifying the establishment of point correspondences between the free form shapes to be matched;

3. It can be justified in the framework of the EM algorithm [26]: in the expectation step, the camera motion parameters are fixed and the correspondence matrix $\mathbf{M}$ is then estimated; in the maximization step, the correspondence matrix $\mathbf{M}$ is fixed and the camera motion parameters rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ are then estimated in the weighted least squares sense; and finally

4. Both the point correspondence probabilities and the camera motion parameters are simultaneously optimised by the efficient deterministic annealing scheme.

As a result, the GA algorithm is theoretically elegant. Practically, once the GA algorithm correctly matches 3D free form shapes with small motions, it can successfully match 3D free form shapes subject to a motion with a rotation angle up to 90° around a certain rotation axis. This robust characteristic implies that, in practice, the number of images to be captured, and thus time required for processing, can be greatly reduced.

However, the GA algorithm has a fatal shortcoming in that its time and space complexities are too high [27] for the matching of 3D free form shapes with thousands of points, which will hinder its application. In this case, resampling [28], feature point extraction and resampling [21], or feature point extraction and fusion [22] has to be used to reduce the number of points for feasible matching. Unfortunately, resampling makes it difficult to replicate the algorithm's performance as different resampling schemes may lead to different results. Since feature extraction is often sensitive to noise due to resampling of points on the object surface, quantization of measurement, shape discontinuity, or different optical characteristics of object surface, feature extraction itself is also a challenging task in both the machine vision and pattern recognition communities. Hence, feature extraction and resampling are just an expedient solution to the problem and they do not really solve the problem, since they transform one difficulty to another.

1.4 Our work

The purpose of this paper is to investigate whether it is indispensable for a point in the first image to match all points in the second as is the case for the GA algorithm. If not, then both the time and space efficiency can be obtained, leading to a promoted application of the GA algorithm for 3D free form shape matching. Hence, we propose using k closest points in the second image for 3D free form shape matching, instead of all points. For the sake of computational efficiency, the optimised k-D tree [29] is employed for the search of the k closest points. Since occlusion and appearance and disappearance of points in different views almost always occur, slack variables have to be employed to explicitly model outliers in the process of matching. The relative quality

of each possible point match is then estimated using the GA algorithm. The motion parameters of interest are finally estimated by the quaternion method [2] in the weighted least squares sense. Both the time and space efficiency enable the proposed improved GA algorithm to operate without any pre-processing directly on 3D free form shapes consisting of either sparse or dense points. The proposed improved GA algorithm has an advantage of easy implementation as it does not require any feature extraction or resampling and it only assumes that the 3D free form shapes to be matched are represented as either sparse or dense unorganised points. This assumption is reasonable since other representation of 3D free form shapes like line segments, triangular meshes, planar patches, or analytic forms can all be transformed into points [2].

For a comparative study of performance, we also implement the original GA algorithm [20] and the geometric ICP algorithm (GICP) [15] with the closest point search accelerated by the optimised k-D tree [29]. The reason why the GICP algorithm was chosen is that it was developed for the matching of unorganised points data and it also has an advantage of easy implementation. While both the GA and improved GA algorithms use the two-way constraint to evaluate the possible point matches, the GICP algorithm uses the rigid motion constraints for the same purpose.

The rest of this paper is structured as follows: Section 2 outlines the GA algorithm, Section 3 extends the GA algorithm while Section 4 presents experimental results. Finally, Section 5 draws some conclusions.

2 Outline of the GA Algorithm

In this paper, the following notations are used: capital letters denote vectors or matrices, lower case letters denote scalars, $||\cdot||$ denotes the Euclidean norm of a vector, superscript T denotes transpose of a vector or a matrix, $|\cdot|$ denotes the absolute value of a scalar, parameters with and without $\hat{}$ in Section 4 denote calibrated and real ones.

Given that the two overlapping free form shapes to be matched are represented as two sets of unorganised points $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_{n_1}\}$ and $\mathbf{P}' = \{\mathbf{p}'_1, \mathbf{p}'_2, \cdots, \mathbf{p}'_{n_2}\}$. Due to occlusion and appearance and disappearance of points, n_1 here is not necessarily equal to n_2. The points with the same subscript do not mean that they are correspondences. The GA algorithm [20] is outlined from the following five aspects: objective function, matching algorithm, property of correspondence matrix, an example of correspondence matrix, and time and space complexities. For details, please refer to [20].

2.1 Objective function

The objective function proposed in [20] for 3D free form shape matching can be rewritten as:

$$E_{3D}(\mathbf{M}, \mathbf{R}, \mathbf{t}) = \sum_{j=1}^{n_2}\sum_{i=1}^{n_1} m_{ij}||\mathbf{p'}_j - \mathbf{R}\mathbf{p}_i - \mathbf{t}||^2 + \frac{1}{\beta}\sum_{j=1}^{n_2}\sum_{i=1}^{n_1} m_{ij}\ln m_{ij} - (\frac{1}{\beta} + \alpha)\sum_{j=1}^{n_2}\sum_{i=1}^{n_1} m_{ij}$$

This objective function simultaneously optimises the correspondence matrix $\mathbf{M} = \{m_{ij}\}$ between $\mathbf{P}$ and $\mathbf{P}'$ to be matched and the camera motion parameters rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$. In this objective function, $m_{ij}(m_{ij} \in [0, 1])$ denotes the probability for the ith point $\mathbf{p}_i$ in $\mathbf{P}$ to match the jth point $\mathbf{p}'_j$ in $\mathbf{P}'$. Intuitively, a point in $\mathbf{P}$ can at most match a point in $\mathbf{P}'$,

and vice versa. Thus, $\sum_{i=1}^{n_1} m_{ij} = 1$ and $\sum_{j=1}^{n_2} m_{ij} = 1$ hold. Since occlusion and appearance and disappearance of points almost always occur in the process of data acquisition, slack variables $\mathbf{p}_{n_1+1}$ and $\mathbf{p}'_{n_2+1}$ have to be introduced so that they can correspond to all occluded and appearing and disappearing points. Consequently, $\sum_{i=1}^{n_1+1} m_{ij} = 1$ and $\sum_{j=1}^{n_2+1} m_{ij} = 1$ then hold. This two way constraint is enforced using the Sinkhorn iterative alternate row and column normalisation procedure [25], leading to an optimal estimation of the correspondence matrix $\mathbf{M}$.

A careful analysis of the objective function $E_{3D}(\mathbf{M}, \mathbf{R}, \mathbf{t})$ above reveals (Figure 1) that the first term is to minimise the matching error of the point match $(\mathbf{p}_i, \mathbf{p}'_j)$. The second term can be rewritten as: $\frac{1}{\beta}\sum_{j=1}^{n_2}\sum_{i=1}^{n_1} m_{ij}\ln m_{ij} = -\frac{1}{\beta}\sum_{j=1}^{n_2}\sum_{i=1}^{n_1}(-m_{ij}\ln m_{ij})$. From information theory [30], it is known that $(-m_{ij}\ln m_{ij})$ denotes the entropy of m_{ij}. When all point matches $(\mathbf{p}_i, \mathbf{p}'_j)$ are equally likely, then $m_{ij} = \frac{1}{n_1 n_2}$ and $Entropy(\mathbf{M}) = \sum_{j=1}^{n_2}\sum_{i=1}^{n_1}(-m_{ij}\ln m_{ij})$ reaches the maximum. For the sake of minimization, the negative of $Entropy(\mathbf{M})$ is used in the objective function $E_{3D}(\mathbf{M}, \mathbf{R}, \mathbf{t})$. Thus, the second term can be interpreted as equalising the weights for different point matches $(\mathbf{p}_i, \mathbf{p}'_j)$, forcing them to yielding the same matching error and consequently, rendering one shape to run parallel, rather than intersecting, with the other due to Equations 1 and 2 in the next section. The third term is to maximise the overlapping area between $\mathbf{P}$ and $\mathbf{P}'$ to be matched. As a result, the objective function $E_{3D}(\mathbf{M}, \mathbf{R}, \mathbf{t})$ can be represented as:

$E = minimizing\ weighted\ matching\ errors + equalizing\ weights + maximizing\ overlapping\ area$

2.2 Matching algorithm

The following algorithm was proposed in [20] to optimise the objective function $E_{3D}(\mathbf{M}, \mathbf{R}, \mathbf{t})$, taking the two-way constraint into account:

Initialize $\mathbf{R}$ to the identity matrix, $\mathbf{t}$, β to β_0, $\hat{m}_{ij}$ to $(1+\epsilon)$

Begin A: Do A until $(\beta \geq \beta_f)$

Begin B: Do B until $\mathbf{M}$ converges or # of iterations $> I_0$

Begin C (**update correspondence parameters by softassign**):

$$Q_{ij} \leftarrow \frac{\partial E_{3D}}{\partial m_{ij}} \tag{1}$$

$$m^0_{ij} \leftarrow exp(-\beta(Q_{ij} - \alpha)) \tag{2}$$

Begin D: Do D until $\hat{\mathbf{M}}$ converges or # of iterations $> I_1$

Update $\hat{\mathbf{M}}$ by nomalising across all rows:

$\hat{m}^1_{ij} \leftarrow \frac{\hat{m}^0_{ij}}{\sum_{j=1}^{j=n_2+1}\hat{m}^0_{ij}}$

Update $\hat{\mathbf{M}}$ by nomalising across all columns:

$\hat{m}^0_{ij} \leftarrow \frac{\hat{m}^1_{ij}}{\sum_{i=1}^{i=n_1+1}\hat{m}^1_{ij}}$

End D

End C

Begin E (**update pose parameters using Walker et al.'s method**):

Update $\mathbf{R}$, $\mathbf{t}$

End E
End B
$\beta \leftarrow \beta_r \beta$
End A
where β_0 denotes the initial temperature control parameter for deterministic annealing, β_r the temperature control parameter increasing rate, β_f the final temperature control parameter, I_0 and I_1 are the maximum iteration numbers, α determines when two points should be regarded as a plausible point match.

2.3 Property of correspondence matrix

The continuous correspondence matrix $\mathbf{M}$ in the GA algorithm converges toward a discrete matrix due to the following two mechanisms that are used concurrently:

1. First, a technique due to Sinkhorn [25] is applied. When each row and column of $\mathbf{M}$ is normalised (several times, alternately) by the sum of the elements of that row and column respectively, the resulting matrix has positive elements with all rows and columns except those corresponding to slack variables summing to one.

2. The term β is increased as the iteration proceeds. As β increases and each row or column of $\mathbf{M}$ is renormalized, the terms m_{ij} corresponding to the smallest Q_{ij} in the overlapping area tend to converge to non-zero and all others except those corresponding to slack variables tend to converge to zero. This is a deterministic annealing process known as softmax. This is desirable behaviour, since it leads to an assignment of point correspondences that satisfy the matching constraints and the minimization of the objective function.

2.4 An example of correspondence matrix

An example of the final correspondence matrix $\mathbf{M}$ estimated by the GA algorithm is presented in Table 1 where data points were generated using the procedure to be described in Section 4. From Table 1, it can be seen that most elements of the correspondence matrix $\mathbf{M}$ are zero, implying that the correspondence matrix $\mathbf{M}$ can be condensed without losing useful information for matching.

From Table 1, it can be seen that $\mathbf{p}_6$ and $\mathbf{p}'_6$ are slack variables, $(\mathbf{p}_2, \mathbf{p}'_1)$, $(\mathbf{p}_3, \mathbf{p}'_2)$, $(\mathbf{p}_4, \mathbf{p}'_3)$, and $(\mathbf{p}_5, \mathbf{p}'_4)$ are real correspondences, $\mathbf{p}_1$ and $\mathbf{p}'_5$ are disappearing and appearing points respectively. The probabilities of real correspondences are not necessarily equal to 1, affected by a number of factors, including β, their actual matching errors, and slack variables.

2.5 Time and space complexities

From the outline of the GA algorithm, it can be seen that the algorithm has both the time and space complexities of $O(n_1 n_2)$ [19]. However, the whole process is often both very time and space consuming [27]. When both n_1 and n_2 are large, the two-way constraint enforcement dominates the matching process. Both the high time and space demands render it difficult, if not impossible, for the GA algorithm to match overlapping 3D shapes with thousands of points. This claim is to be demonstrated in Section 4.

3 The improvement to GA

As outlined in the last section, the GA algorithm is essentially time consuming and storage expensive [27]. On the other hand, while the GA algorithm uses all points for matching, we wonder whether it is absolutely necessary for each point in one image to match all points in another. This motivates us to propose using k-closest points instead for matching. In this case, the rows in correspondence matrix $\mathbf{M}$ corresponding to points in the first image and slack variable $\mathbf{p}_{n_1+1}$ have different numbers of columns. For the sake of avoiding a misuse of term *matrix*, the correspondence matrix $\mathbf{M}$ is thus called ragged matching array. The proposed improvement to the GA algorithm is made from the following five aspects: ragged matching array computation, the two-way constraint enforcement, camera motion estimation, an example of ragged matching array, and time and space complexities which are detailed as follows.

3.1 Ragged matching array computation

For each point $\mathbf{p}_i$ in $\mathbf{P}$, the optimised k-D tree [29] can be used to find k closest points $\mathbf{p}'_{c_{i1}}$, $\mathbf{p}'_{c_{i2}}$, $\cdots$, and $\mathbf{p}'_{c_{ik}}$ in $\mathbf{P}'$ to the transformed point $\mathbf{R}\mathbf{p}_i + \mathbf{t}$. Here, $c_{ij}(j = 1, 2, \cdots, k)$ denotes the subscript of points in $\mathbf{P}'$. The operation of finding k closest points here is essentially a mapping C that associates k closest points in $\mathbf{P}'$ to each point $\mathbf{p}_i$ in $\mathbf{P}$. This mapping C plays a critical role for the computational efficiency of the proposed improved GA algorithm. Then the corresponding matching error Q can be computed as:

$$Q_{ij} = ||\mathbf{p}'_{c_{ij}} - \mathbf{R}\mathbf{p}_i - \mathbf{t}||^2 \ (i = 1, 2, \cdots, n_1; j = 1, 2, \cdots, k),$$
$$Q_{ik+1} = ||\mathbf{p}'_{n_2+1} - \mathbf{R}\mathbf{p}_i - \mathbf{t}||^2 \ (i = 1, 2, \cdots, n_1), \quad Q_{n_1+1j} = ||\mathbf{p}'_j - \mathbf{R}\mathbf{p}_{n_1+1} - \mathbf{t}||^2 \ (j = 1, 2, \cdots, n_2).$$

Consequently, the ragged matching array $\mathbf{M}$ can be computed as:

$$m_{ij} = exp(-\beta(Q_{ij} - \alpha)) \ (i = 1, 2, \cdots, n_1; j = 1, 2, \cdots, k),$$
$$m_{ik+1} = exp(-\beta_0(Q_{ij} - \alpha)) \ (i = 1, 2, \cdots, n_1), \quad m_{n_1+1j} = exp(-\beta_0(Q_{ij} - \alpha)) \ (j = 1, 2, \cdots, n_2).$$

The ragged matching array $\mathbf{M}$ has the following structure:

$$\begin{pmatrix} m_{11} & m_{12} & \cdots & m_{1k} & m_{1k+1} \\ m_{21} & m_{22} & \cdots & m_{2k} & m_{2k+1} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ m_{n_1 1} & m_{n_1 2} & \cdots & m_{n_1 k} & m_{n_1 k+1} \\ m_{n_1+11} & m_{n_1+12} & \cdots\cdots & m_{n_1+1 n_2} & \end{pmatrix}.$$

3.2 The two-way constraint enforcement

For the application of the Sinkhorn procedure for imposing the two-way constraint, special attention must be given. For each row i ($i = 1, 2, \cdots, n_1$), the normalisation is relatively easy to implement: $\hat{m}^1_{ij} \leftarrow \frac{\hat{m}^0_{ij}}{\sum_{j=1}^{j=k+1} \hat{m}^0_{ij}}$. However, the column normalisation is significantly different. Even though the former n_1 rows have just k columns respectively, we have to process them in the same manner as the last row, corresponding to each point in $\mathbf{P}'$. The column normalisation can be implemented in the following three steps as:

1. Initialise the sum of matching probabilities for each point in $\mathbf{P}'$ as: $s(j) = \hat{m}^1_{n_1+1j}$ $(j = 1, 2, \cdots, n_2)$;

2. Consider the matching probability for any point in $\mathbf{P}'$ found as a possible point correspondent by a point in $\mathbf{P}$:

$$s(c_{ij}) \leftarrow s(c_{ij}) + \hat{m}^1_{ij} \ (i = 1, 2, \cdots, n_1; j = 1, 2, \cdots, k);$$

3. Normalise each column (with sparse elements) in $\mathbf{M}$ as:

$$\hat{m}^0_{ij} = \frac{\hat{m}^1_{ij}}{s(c_{ij})} \ (j = 1, 2, \cdots, k; i = 1, 2, \cdots, n_1), \quad \hat{m}^0_{n_1+1j} = \frac{\hat{m}^1_{n_1+1j}}{s(j)} \ (j = 1, 2, \cdots, n_2).$$

The two-way constraint enforcement leads to an optimal estimation of the ragged matching array $\mathbf{M}$.

3.3 Camera motion estimation

Once the ragged matching array $\mathbf{M}$ has been estimated, the motion parameters rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ can be estimated using the quaternion method [2] in the weighted least squares sense as:

1. Initialise the matching probabilities w_i $(i = 1, 2, \cdots, n_1)$ and w'_j $(j = 1, 2, \cdots, n_2)$ as zero for each point in $\mathbf{P}$ and $\mathbf{P}'$ respectively;

2. Compute the accumulated matching probabilities w_i and w'_j for each point in $\mathbf{P}$ and $\mathbf{P}'$ respectively as:

$$w_i \leftarrow w_i + m_{ij} \ (j = 1, 2, \cdots, k), \quad w'_{c_{ij}} \leftarrow w'_{c_{ij}} + m_{ij} \ (i = 1, 2, \cdots, n_1; j = 1, 2, \cdots, k);$$

3. Compute the weighted centroid of each point set as: $\bar{\mathbf{p}} = \frac{1}{\sum_{i=1}^{n_1} w_i} \sum_{i=1}^{n_1} w_i \mathbf{p}_i$, $\bar{\mathbf{p}}' = \frac{1}{\sum_{j=1}^{n_2} w'_j} \sum_{j=1}^{n_2} w'_j \mathbf{p}'_j$;

4. Compute the covariance matrix $\mathbf{H}$ as: $\mathbf{H} = \sum_{i=1}^{n_1} \sum_{j=1}^{k} m_{ij} (\mathbf{p}_i - \bar{\mathbf{p}})(\mathbf{p}'_{c_{ij}} - \bar{\mathbf{p}}')^T$;

5. Use the quaternion method [2] to estimate the rotation matrix $\mathbf{R}$ from $\mathbf{H}$, leading the translation vector $\mathbf{t}$ to be estimated in the weighted least squares sense as: $\mathbf{t} = \bar{\mathbf{p}}' - \mathbf{R}\bar{\mathbf{p}}$.

Then the above steps in the traditional GA algorithm described in the last section can be iterated. When the relative variation of either the rotation or translation vector at successive two iterations is smaller than the desired matching error ρ or the iteration number is larger than I_1, the cycle $\mathbf{B}$ terminates. In practice, the algorithm is not sensitive to ρ. In general, it can be set in the interval [0.00001, 0.1].

3.4 An example of ragged matching array

An example of the final ragged matching array $\mathbf{M}$ estimated by the proposed improved GA algorithm is presented in Table 2 where data points were generated using the procedure to be described in Section 4. From Table 2, it can be seen that $\mathbf{M}$ here still possesses the properties of the correspondence matrix described in the last section and correctly identifies all the real correspondences and appearing and disappearing points between the two overlapping 3D free form shapes to be matched. However, it is much more compact than that in Table 1. This shows that condensing the correspondence matrix without losing useful information for matching is feasible.

3.5 Time and space complexities

From the development of the proposed improved GA algorithm, it can be seen that it has a time complexity of $O(n_1 k \ln n_2)$ and a space complexity of $O(n_1 + n_2)$, which are much smaller than those, $O(n_1 n_2)$, of the traditional GA algorithm. Both the time and space efficiency enable the improved GA algorithm to operate without any pre-processing directly on whole 3D free form shapes consisting of either sparse or dense points as demonstrated in the next section. Since the proposed improvement to the GA algorithm is based on k closest points, we call it the k-GA algorithm.

4 Experimental results

In order to provide a better understanding of the performance of the improved k-GA algorithm, we implemented the original GA algorithm, the improved k-GA algorithm and the GICP algorithm [15] with the desired matching error equal to 0.003 and the maximum iteration number equal to 300 for a comparative study based on both synthetic data and real images. All algorithms were directly applied to data points without any pre-processing, feature extraction, or segmentation and also without any knowledge about the distribution of points, occlusion, appearance and disappearance of points, or motion information. Thus, the experiments based on such images are objective and they represent typical imaging conditions.

All these algorithms were initialised using a pure translational motion derived from the centroid difference of the first and second point sets, representing the free form shapes to be matched. In addition, the following parameter values were used: slack variables $\mathbf{p}_{n_1+1} = \frac{1}{n_1}\sum_{i=1}^{n_1}\mathbf{p}_i$ and $\mathbf{p}'_{n_2+1} = \frac{1}{n_2}\sum_{j=1}^{n_2}\mathbf{p}'_j$, $\bar{d} = \frac{1}{n_1 n_2}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}||\mathbf{p}'_j - \mathbf{p}_i||^2$, $\alpha = 0.03$, $\beta_0 = 0.1/\bar{d}$, $\beta_r = 1.1$, $\beta_f = 4000/\bar{d}$, $I_0 = 30$, $I_1 = 10$, $\rho = 0.001$, and $\hat{\mathbf{M}}$ convergence threshold $\gamma = 0.05$.

The parameters of interest in this paper are the number k of closest points for matching, the relative calibration error in percentage of rotation axis as $e_{\mathbf{h}} = ||\hat{\mathbf{h}} - \mathbf{h}||$, the relative calibration error in percentage of rotation angle as $e_\theta = (\hat{\theta} - \theta)/\theta$, and the relative calibration error in percentage of translation vector as $e_{\mathbf{t}} = ||\hat{\mathbf{t}} - \mathbf{t}||/||\mathbf{t}||$ for the synthetic points data, the average e_μ and standard deviation e_σ in millimetres of matching errors e_i based on the finally established N reciprocal correspondences [5, 31] $(\mathbf{r}_i, \mathbf{r}'_i)(i = 1, 2, \cdots, N)$: $e_i = ||\mathbf{r}'_i - \hat{\mathbf{R}}\mathbf{r}_i - \hat{\mathbf{t}}||$, $e_\mu = \frac{1}{N}\sum_{i=1}^{N} e_i$, and $e_\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(e_i - e_\mu)^2}$, the calibrated rotation angle $\hat{\theta}$ in degrees: $\hat{\theta} = \frac{180}{\pi} acos \frac{r_{11}+r_{22}+r_{33}-1}{2}$,

$\hat{\mathbf{h}} = (\frac{r_{23}-r_{32}}{2\sin(\hat{\theta})}, \frac{r_{31}-r_{13}}{2\sin(\hat{\theta})}, \frac{r_{12}-r_{21}}{2\sin(\hat{\theta})})^T$, $\hat{\mathbf{R}} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$, and the matching time in seconds for real images. Since the rotation angle θ around an unknown rotation axis of the camera motion can be derived from the name encoding of real image files, the calibrated rotation angle $\hat{\theta}$ is thus in this paper one of the parameters of interest. All the experiments were carried out on a Pentium III, 866MHz, 256M RAM computer.

4.1 Synthetic data with sparse points

First n points $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n\}$ were randomly generated with uniform distribution within the 3D space $[10, 20] \times [10, 20] \times [10, 20]$. These points were then subjected to a rotation angle θ around a fixed rotation axis $\mathbf{h}$ randomly generated with uniform distribution within the 3D space $[1, 3] \times [1, 3] \times [1, 3]$ followed by a constant translation vector $\mathbf{t}$ randomly generated with uniform distribution within the 3D space $[10, 20] \times [10, 20] \times [10, 20]$. Let the transformed points be $\mathbf{P}' = \{\mathbf{p}'_1, \mathbf{p}'_2, \cdots, \mathbf{p}'_n\}$. Once the data were generated we thus, have precise knowledge of the selected points and their correspondences $(\mathbf{p}_i, \mathbf{p}'_i)(i = 1, 2, \cdots, n)$ and motion parameters rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ to serve as reference for error estimation and validation of the algorithms.

In order to simulate real world noise contaminated data, Gaussian random noise was added to the coordinates of each point with mean equal to 0 and standard deviation $\sigma_1 = 0.1$ in one series of experiments and $\sigma_2 = 0.2$ in another. In order to simulate occlusion and appearance and disappearance of points, we removed the last 20% points in $\mathbf{P}$ and the first 20% points in $\mathbf{P}'$. Finally we obtained two new sets of points $\mathbf{P}$ and $\mathbf{P}'$ for matching with 60% overlapping in 3D space.

4.1.1 The number k of closest points

In this section, we report the experimental results about how to determine the optimal number k ($1 \leq k \leq n_2$) of closest points for matching. Unless stated elsewhere, the synthetic data points used for experiments are always corrupted by the low level σ_1 of noise as described above. The experimental results are presented in Table 3.

From Table 3, it can be seen that as pointed out in [20], the GA algorithm does sometimes converge to a local minimum, leading to poor matching results. Unfortunately, the k-GA algorithm also cannot guarantee to always converge to the global minimum. If the number k of closest points is too small, then a limited number of candidate points in the second view are chosen. In this case, the k-GA algorithm does not make full use of information in the shape for matching. In addition, the smaller the parameter k is, the more heavily the order of images biases the final matching results. If the number k of closest points is too large, then too many candidate points in the second view chosen for matching not only will require more time for matching, but also may confuse which is the best match. This is especially the case when the 3D shapes are represented as sparse points. This observation is justified by psychological studies as follows: the probability for a stimulus to confuse another decreases exponentially with the distance between these two stimuli [32]. This explains that the points farther from the closest points in the second image are unlikely to ambiguate the 3D free form shape matching and thus, a limited number k of closest points maybe enough for accurate 3D free form shape matching. Moreover, when

human brain responds to a stimulus, it inhibits the response to other stimuli [33]. This shows that the response from the human brain to the closest points may inhibit the response to those farther away from the closest points. Thus, considering k closest points in the second view for matching, instead of all points, is both practically and theoretically feasible. An overall analysis of the experimental results reveals that k=4 is a good compromise for matching, achieving both accuracy and stability. Thus, unless stated elsewhere, k=4 is always used for matching in the rest of this paper.

From Table 3, it can also be seen that sometimes, the final matching results are not dependent on the number k of closest points. This can be explained as follows: for any point $\mathbf{p}_i$ in $\mathbf{P}$, we establish k possible point matches $(\mathbf{p}_i, \mathbf{p}'_{c_{ij}})$ $(j = 1, 2, \cdots, k)$ and at most one of them is real. If $\mathbf{p}_i$ is in the overlapping area, then one of them must be somewhat feasible at different evolutionary stages of matching. If $\mathbf{p}_i$ is a disappearing point, then none of them will be feasible. Whether the real point correspondences can be finally established or not is largely determined by the configuration of points, the characteristics of the shapes to be matched, and the motion that the camera underwent. This phenomenon is not dependent on the number k of closest points and is reflected as a fact that different numbers k of closest points are likely to lead the k-GA algorithm to converge to the same local minimum, yielding the same final matching results. For accurate matching, the k-GA algorithm has to converge to the global minimum, independent of the size of k. However, in practice, various numbers k of closest points may result in paths with different efficiencies and/or probabilities for successfully evolving the matching process of establishing real point correspondences between $\mathbf{P}$ and $\mathbf{P}'$ to be matched.

When k=1, the k-GA algorithm degenerates to the traditional ICP algorithm. In this case, we call the k-GA algorithm the SoftICP algorithm. The main difference between the SoftICP and ICP algorithms lies in that the former imposes the two-way constraint embedded into a stochastic optimisation scheme in the form of deterministic annealing, while the latter does not.

4.1.2 Different motions

In this section, we do a comparative study of performance among the GA, k-GA, and GICP algorithms. The experimental results are presented in Figure 2 and Table 4 (n=90). In the figure, the solid lines correspond to the low level σ_1 of noise, the dash lines correspond to the high level σ_2 of noise, lines with pluses correspond to the k-GA algorithm, lines with crosses correspond to the GICP algorithm, and lines without any signs correspond to the GA algorithm.

From Figure 2, it can be seen that while both the GA and k-GA algorithms are significantly more accurate than the GICP algorithm, the GICP algorithm is not stable. This is because the GICP algorithm applied the Monte Carlo resampling technique to estimate some motion parameters of interest necessary for the elimination of false matches and differently resampled data may lead to different motion estimation results. With the rotation angle increasing, both the GA and k-GA algorithms become more and more accurate for the calibration of the parameters of interest rotation axis, rotation angle, and translation vector. This shows that small motions are relatively difficult to calibrate. When data points were corrupted by heavier noise, the performance of both the GA and k-GA algorithms is poorer. This however is expected, since heavier noise makes data points more unreliable.

Even though all points in the second view are used for matching, the GA algorithm still yielded very similar results to those by the k-GA algorithm for the calibration of the parameters of

interest rotation axis, rotation angle and translation vector. This shows that it is not necessary to use all points in $\mathbf{P}'$ for matching. However, the k-GA algorithm is about 20 times faster and thus in this case is significantly more computationally efficient than the GA algorithm. This conclusion is confirmed by Table 4. This justifies our motivation to look at k closest points in the second view for matching, instead of all points.

4.1.3 Different numbers of points

In this section, we report the experimental results about the relationship between the number of points on the 3D free form shapes to be matched and the time required for matching. The rotation angle of the camera motion was fixed: $\theta = 25°$. The experimental results are presented in Figure 3 and Table 5.

From Figure 3, it can be seen that all algorithms fluctuate with respect to the calibration of the parameters of interest rotation axis, rotation angle and translation vector. This may be because the same level (σ_1) of noise has different effects in corrupting different numbers of randomly generated points with uniform distribution on the 3D free form shapes to be matched, leading to different accuracies for the calibration of the parameters of interest. But a careful analysis still reveals that both the GA and k-GA algorithms are more stable than the GICP algorithm.

While both the GA and k-GA algorithms again produce similar matching results for the calibration of the parameters of interest rotation axis, rotation angle, and translation vector, the time required by the GA algorithm increases rapidly with the number of points on the 3D free form shapes to be matched. A careful analysis confirms that the GA algorithm has a time complexity of $O(n_1 n_2)$ and the relationship between the time required for matching and the numbers n_1 and n_2 of points on the 3D free form shapes to be matched can be approximated as: time(s)=4*n_1/40* n_2/40. Based on this approximation, we can predict that matching two images, where each is composed of 8000 points, as is the case described in the next section, will take about 4*8000/40*8000/40=160,000 seconds. This is really a long time for the matching of two typical and relatively small, whole real images. This confirms the claim described in [27] and in Section 2 of this paper and explains why the traditional GA algorithm has to employ various techniques to reduce the number of points for feasible matching. However, the k-GA algorithm increases time smoothly for matching. This is a desired characteristic for matching algorithms to deal with 3D free form shapes with various sizes.

4.2 Real images with dense points

In this section, we report the experimental results based on real images. Since k closest points in the second image, instead of all points, are used for matching by the k-GA algorithm, the concern whether the order of images effects the k-GA algorithm on the final matching results arises. To deal with this concern, we did experiments using both orders of real images. Since the GA algorithm is intolerably slow in processing the whole images with dense points, we implemented it with a uniform resampling of image points at a rate of 1/10 so that about 10% points were selected. Of course, the experiments based on resampling are not objective and maybe unfair. Our interest however is just to provide a rough idea about how the resampling scheme influences the performance of the traditional GA algorithm. To avoid giving a wrong impression about the performance of the GA algorithm, only two parameters are recorded: one is the calibrated

rotation angle $\hat{\theta}$ of the camera motion, which should be relatively independent of resampling with the assumption that the resampled points can approximate the 3D free form shapes to be matched, the other is the time used for matching from which we can predict how much time is needed for the matching of whole images. Note that while the GA algorithm was applied to sampled points for matching, both the k-GA and GICP algorithms were applied to all image points.

The range images depicted in Figures 4 and 6 and and Tables 6 and 9 used here were downloaded from the range image database currently hosted at the Signal Analysis and Machine Perception Laboratory at Ohio State University. All range images were captured using a Minolta Vivid 700 range camera and are of the same size of 200×200 pixels.

For a better visualisation of 3D free form shape matching, 200 points were randomly selected with uniform distribution from the first images and their matching results are presented below. In the figures, pluses represent the transformed first image points, circles represent the second image points closest to the transformed first image points.

4.2.1 Small motions

Since the GICP algorithm requires a good initialisation of motion parameters, in this section, we report the experimental results for the automatic matching of 3D free form shapes subject to relatively small motions. By doing this, we provide an ideal condition for all algorithms to match 3D free form shapes. The images chosen are bird and tubby (Figure 4 and Table 6). The experimental results are presented in Figure 5 and Tables 7 and 8.

From Figure 5, it can be seen that before matching, the two sets of points are significantly different in 3D space. But after matching, the two sets of points are perfectly superimposed and appearing and disappearing points have been correctly identified. This observation is clearly confirmed by Table 7 where before matching, the average and standard deviation of matching errors based on reciprocal correspondences are large. After matching these parameters of interest have been significantly reduced. This visualisation clearly shows that the proposed algorithm is accurate for 3D free form shape matching.

From Table 8, it can be seen that while the k-GA algorithm achieves a matching accuracy of about 1/3 interpoint distance in images in Table 6, it is significantly more accurate than the GICP algorithm in the sense of average matching error over reciprocal correspondences. The GA algorithm performs really poorly especially for the matching of the tubby images. While the k-GA algorithm calibrated a rotation angle of 19.70° of the camera motion for the matching of the tubby2 and tubby1 images, the GA algorithm calibrated the corresponding rotation angle as 3.52°, which is completely inaccurate and far from the expected 20°.

While the order of images used for matching imposes a great effect on the GICP algorithm with respect to the final matching results, it does not really bother the k-GA algorithm for the automatic 3D free form shape matching. For example, while different orders of two tubby images lead the GICP algorithm to yield a relative average matching error variation of $2|0.38 - 0.26|/(0.38+0.26)*100\%=37.50\%$, they do not really lead the k-GA algorithm to yield a significant difference between the corresponding average matching errors. This is because the k-GA algorithm applies the two-way constraint, reducing the effect of the order of images on the final matching results. When the first image is larger than the second image as is the case for the matching of the bird2 and bird1 images, more false matches were introduced, leading the GICP

algorithm to converge to a local minimum with a relatively larger matching error of 0.39mm.

From the time complexity of the GA algorithm, we can speculate again that it will take about 19* 9234/112 *10502/128 = 128,524 seconds for the matching of the whole bird images and 4* 4354/55 * 4361/57 = 24,226 seconds for the matching of the whole tubby images. The time predicted here is less than that in the last section. This is because the GA algorithm converges prematurely here for the matching of the bird and tubby images with resampling, yielding poor matching results. In this case, even though the time is tolerable, the space demand is still too high, since we can hardly open an array of 9234 by 10502 on a PC. The k-GA algorithm is less computationally efficient than the GICP algorithm. This is because the latter did not completely eliminate false matches, accelerating the matching process [16].

4.2.2 Large motions

In this section, we report the experimental results for the automatic matching of 3D free form shapes subject to relatively large motions with a large number of occluded and appearing and disappearing points. Since large motions violate the assumption of the GICP algorithm, it is expected that the GICP algorithm will perform poorly. However, our interest is to test how the k-GA algorithm behaves in dealing with large motions. The images chosen are bunny and dinosaur (Figure 6 and Table 9). The experimental results are presented in Figure 5 and Tables 7 and 10.

From Figure 5 and Table 7, it can be seen that even for images with relatively large motions, the proposed algorithm can still accurately match them despite the fact that only 4 closest points in the second image were used for matching. This shows that the proposed algorithm is powerful for automatically matching images with a large range of motions without loss of accuracy.

From Table 10, it can be seen that while both the GA and GICP algorithms perform reasonably well with all relative calibration errors of rotation angle less than 37.58%, the k-GA algorithm is again significantly more accurate than the GICP algorithm. Also, the order of images imposes a subtle effect for matching on the GICP algorithm with respect to the final matching results, depending on the specific 3D free form shapes to be matched. However, the k-GA algorithm yields relatively stable results, similar to those for small motions. This shows that the k-GA algorithm is powerful in dealing with the automatic matching of 3D free form shapes with various motions. Comparing Table 8 with Table 10, it can be clearly seen that resampling makes the GA algorithm unstable for matching. This confirms our analysis in the introduction.

5 Conclusions

In this paper, we have proposed using k closest points for matching with the point match probabilities estimated using the graduated assignment algorithm, leading the camera motion parameters to be estimated using the quaternion algorithm in the weighted least squares sense. Since outliers almost always occur in the process of 3D free form shape matching, slack variables were employed to explicitly model them. In enforcing the two-way constraint using the Sinkhorn procedure, while the row normalisation is relatively easy to implement, the column normalisation needs special attention: even though a row has just k columns, we have to process them as n_2 columns, corresponding to each point in the second view. Comparing the k-GA algorithm with the GA algorithm, we can draw the following conclusions:

- While the latter uses all points in the second view for matching and thus, has high time and space demands, the former uses k closest points. Since k is generally considerably smaller than the number of points in the second view, it dramatically reduces both the time and space complexities of the latter. However, the experimental results based on both synthetic data and real images have shown that the former does not significantly sacrifice the accuracy of the latter for matching and the resulting order of images does not significantly bias the proposed k-GA algorithm on the final matching results;

- Both the time and space efficiency enable the former to successfully operate without any pre-processing directly on whole images with thousands of points, while the latter has difficulty, if does not find it impossible, due to the high time and space demands [27];

- While resampling [28], feature point extraction and resampling [21], or feature point extraction and fusion [22] is often used to reduce the number of points for the latter to match 3D free form shapes with both the time and space feasibility, the optimised k-D tree [29] can be employed to accelerate the former; and finally,

- While feature extraction [14, 21, 22] itself is also a challenging problem in both the machine vision and pattern recognition literatures, the former does not require to extract any features and thus, has an advantage of easy implementation.

Comparing the k-GA algorithm with the GICP algorithm, we can draw the following conclusions:

- The former is often significantly more accurate and stable than the latter, while the latter is significantly more computationally efficient than the former, as the latter does not always completely succeed in eliminating false matches, speeding up the matching process [16];

- The order of images imposes a significantly heavier impact on the latter for matching than on the former.

Thus, the k-GA algorithm combines the advantages of accuracy of the traditional GA algorithm and both the time and space efficiency of the traditional ICP algorithm and thus, it is powerful and of general use for the automatic matching of 3D free form shapes with either sparse or dense points. The experimental results based on both synthetic data and real images have shown that the proposed k-GA algorithm yields very stable results for the automatic matching of 3D free form shapes despite the fact that it was just initialised by a pure translational motion derived from the centroid difference of the two shapes to be matched.

One of the most important findings described in this paper is that when the 3D free form shapes to be matched are represented as either sparse or dense unorganised points and subjected to a relatively small motion with a rotation angle smaller than 30°, for instance, it is the two-way constraint embedded into a stochastic optimisation scheme in the form of deterministic annealing, not a large number of closest points, that plays a vital role for accurate 3D free form shape matching, more closest points used for matching do not necessarily yield significantly better matching results, but often require significantly more time. Further research is to investigate how to further improve the computational efficiency of the k-GA algorithm. Research is under way and the results will be reported in the future.

6 Summary

The GA algorithm is an elegant algorithm for the automatic matching of 3D shapes. However, its high time and space complexities make it difficult, if not impossible, to operate on whole images with dense points representing 3D shapes to be matched. Thus, feature point extraction or resampling becomes the only feasible choice in this case so that both the time and space complexities of the GA algorithm are tolerable for the matching of 3D shapes with thousands of points. Unfortunately, feature point extraction and resampling have two shortcomings: firstly, it is difficult to replicate the results since different resampling schemes may lead to different results; secondly, when feature points are extracted and resampled, the difficulty in feature extraction is then introduced.

In this paper, we have investigated whether it is necessary for a point in one image to match all points in another as is the case for the traditional GA algorithm. If not, then both the time and space efficiency can be obtained and thus, promote the application of the GA algorithm. This motivated us to propose using k closest points in the second view for matching, instead of all points. In order to improve the computational efficiency of the proposed improved algorithm, the optimised k-D tree is employed. To deal with occlusion and appearance and disappearance of points, slack variables are employed. The probability of each point match is then estimated using the GA algorithm, resulting in the motion parameters to be estimated using the quaternion method in the weighted least squares sense. In applying the Sinkhorn procedure for enforcing the two-way constraint [25], the row normalisation is relatively easy to implement. However, the column normalisation needs special attention. Even though a row has just k columns, we have to treat them as n_2 columns, corresponding to each point in the second view.

The experimental results based on both synthetic data and real images have shown that the proposed improved k-GA algorithm does not significantly sacrifice the accuracy of the traditional GA algorithm and the order of images does not significantly bias the proposed k-GA algorithm on the final matching results, but it gains without any pre-processing both the time and space efficiency for the automatic matching of 3D free form shapes with thousands of points. While it may not be necessary to use all points in the second view as candidates for matching, it is vital to enforce the two-way constraint embedded into a stochastic optimisation scheme in the form of deterministic annealing.

Acknowledgements

We would like to express our sincere thanks to Jo Eastlake from the University of Wales at Aberystwyth for her proofreading of the paper. Thanks also go to the anonymous reviewer whose valuable comments have improved the presentation quality of the paper.

References

[1] Z. Lin, H. Lee, and T. Huang, Finding 3D point correspondences in motion estimation. *Proc. 8th ICPR*, 1986, pp. 303-305.

[2] P. J. Besl, N. D. McKay, A method for registration of 3D shapes. *IEEE Trans. PAMI* 14(1992) 239-256.

[3] Y. Chen and G. Medioni, Object modelling by registration of multiple range images, *IVC* 10(1992) 145-155.

[4] Z. Zhang, Iterative point matching for registration of free-form curves, *Technical report*, no. 1658, INRIA, France, May 1992.

[5] Y. Liu, Improving ICP with easy implementation for free form surface matching. *Pattern Recognition* 37(2004) 211-226.

[6] Y. Liu, L. Li, and B. Wei, 3D shape matching using collinearity constraint. *Proceedings of IEEE 2004 International Conference on Robotics and Automation*, April 26-1 May, 2004, New Orleans, LA, USA, vol. III, pp. 2285-2290.

[7] Y. Liu and B. Wei, Evaluating structural constraints for accurate range image registration. *Proceedings of The 4th International Conference on 3-D Digital Imaging and Modeling(3DIM'03)*, 2003, pp. 187-194.

[8] M.A. Rodrigues and Y. Liu, On the representation of rigid body transformations for accurate registration of free form shapes, *Robotics and Autonomous Systems* 39(1)(2002) 37-52.

[9] D. Hahnel, S. Thrun, and W. Burgard, An extension of the ICP algorithm for modelling nonrigid objects with mobile robots. *Proc. IJCAI*, 2003, pp. 915-920.

[10] C. Dorai, G. Wang, A.K. Jain, and C. Mercer, From images to models: automatic model construction from multiple views. *Proc. ICPR*, 1996, pp. 770-774.

[11] A.E. Johnson and S.B. Kang, Registration and integration of textured 3-D data, *IVC* 17(1999) 135-147.

[12] G.C. Sharp, S.W. Lee, and W.K. Wehe, Invariant features and the registration of rigid bodies, *IEEE Trans. PAMI* 24(2002) 90-112.

[13] S.M. Yamany and A.A. Farag, Surface signatures: an orientation independent free-form surface representation scheme for the purpose of objects registration and matching, *IEEE Trans. PAMI* 24(2002) 1105-1120.

[14] D. Huber, M. Hebert, Fully automatic registration of multiple 3D data sets, *IVC* 21(2003) 637-650.

[15] Y. Liu, M. A. Rodrigues, and Ying Wang. Developing rigid motion constraints for the registration of free-form shapes. *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2000, pp. 2280-2285.

[16] Y. Liu and B. Wei, Developing structural constraints for accurate registration of overlapping range images, *Robotics and Autonomous Systems* 47(2004) 11-30.

[17] R. T. Whitaker, et al, Indoor scene reconstruction from sets of noisy range images. *Proc. 3DIM*, 1999, pp. 348-357.

[18] A. P. Ashbrook, R. B. Fisher, et al., Finding surface correspondences for object recognition and registration using pair-wise geometric histogram. *Proc. 5th ECCV*, 1998, vol. II, pp. 185-201.

[19] S. Gold and A. Rangarajan, A graduated assignment algorithm for graph matching, *IEEE Trans. PAMI* 18(1996) 377-388.

[20] S. Gold, A. Rangarajan, et al, New algorithms for 2-D and 3-D point matching: pose estimation and correspondence, *Pattern Recognition* 31(1998) 1019-1031.

[21] H. Chui and A. Rangarajan, A new point matching algorithm for non-rigid registration, *Computer Vision and Image Understanding* 89(2003) 114-141.

[22] H. Chui and A. Rangarajan, A unified non-rigid feature registration method for brain mapping, *Medical Image Analysis* 7(2003) 113-130.

[23] P. David, D.DeMenthon, R. Duraiswami, and H. Samet, Simultaneous pose and correspondence determination using line features. *Proc. CVPR*, 2003, vol. II, pp. 423-431.

[24] E.T. Jaynes, Information theory and statistical mechanics, *The Physical Review* 106(1957) 620-630.

[25] R. Sinkhorn, A relationship between arbitrary positive matrices and doubly stochastic matrices, *Ann. Math. Statist.* 35(1964) 876-879.

[26] A.P. Dempster, N.M. Laird, and D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *J. Royal Statistical Society Series B* 39(1977) 1-38.

[27] H. Jonsson and B. Soderberg, Deterministic annealing and nonlinear assignment. *Technical report 01-16*, Department of Theoretical Physics, Lund University, 2001.

[28] Y. Liu, L. Li, and Y. Wang, Free form shape matching using deterministic annealing and softassign. *Proc. ICPR*, 2004, vol. II, pp. 128-131.

[29] J. H.Friedman, J. L. Bently, and P. A. Finkel, An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Soft.* 3(1977) 209-226.

[30] C.F. Shannon, A mathematical theory of communication, *Bell Systems Technical Journal* 27(1948) 379-423.

[31] T. Pajdla, L. Van Gool, Registration of 3-D curves using semi-differential invariants. *Proc. ICCV*, 1995, pp. 390-395.

[32] R.N. Shepard, Toward a universal law of generalisation for psychological science, *Science* 237(1987) 1317-1323.

[33] L. Tamm, V. Menon, and A.L. Reiss, Maturation of brain function associated with response inhibition, *J. Am. Acad. Child Adolesc. Psychiatry* 41(2002) 1231-1238.

Table 1: The final correspondence matrix $\mathbf{M}$ estimated by the GA algorithm using synthetic points data (n=6) with $\theta = 25^\circ$ to be described in Section 4.

	$\mathbf{p}'_1$	$\mathbf{p}'_2$	$\mathbf{p}'_3$	$\mathbf{p}'_4$	$\mathbf{p}'_5$	$\mathbf{p}'_6$
$\mathbf{p}_1$	0.000	0.000	0.000	0.000	0.000	1.000
$\mathbf{p}_2$	0.407	0.000	0.000	0.000	0.000	0.593
$\mathbf{p}_3$	0.000	0.410	0.000	0.000	0.000	0.590
$\mathbf{p}_4$	0.000	0.000	0.413	0.000	0.000	0.587
$\mathbf{p}_5$	0.000	0.000	0.000	0.412	0.000	0.588
$\mathbf{p}_6$	0.593	0.590	0.587	0.588	1.000	

Table 2: The final ragged matching array $\mathbf{M}$ estimated by the proposed improved GA algorithm using synthetic points data (n=6 and k=4) with $\theta = 25^\circ$ to be described in Section 4.

$\mathbf{p}_1$	$\mathbf{p}'_3$	$\mathbf{p}'_2$	$\mathbf{p}'_4$	$\mathbf{p}'_1$	$\mathbf{p}'_6$
	0.000	0.000	0.000	0.000	1.000
$\mathbf{p}_2$	$\mathbf{p}'_1$	$\mathbf{p}'_5$	$\mathbf{p}'_4$	$\mathbf{p}'_2$	
	0.376	0.000	0.000	0.000	0.624
$\mathbf{p}_3$	$\mathbf{p}'_3$	$\mathbf{p}'_2$	$\mathbf{p}'_1$	$\mathbf{p}'_4$	
	0.000	0.379	0.000	0.000	0.621
$\mathbf{p}_4$	$\mathbf{p}'_3$	$\mathbf{p}'_2$	$\mathbf{p}'_5$	$\mathbf{p}'_1$	
	0.382	0.000	0.000	0.000	0.618
$\mathbf{p}_5$	$\mathbf{p}'_4$	$\mathbf{p}'_5$	$\mathbf{p}'_1$	$\mathbf{p}'_3$	
	0.381	0.000	0.000	0.000	0.619
$\mathbf{p}_6$	$\mathbf{p}'_1$	$\mathbf{p}'_2$	$\mathbf{p}'_3$	$\mathbf{p}'_4$	$\mathbf{p}'_5$
	0.624	0.621	0.618	0.619	1.000

Table 3: The mean μ and standard deviation σ of relative calibration errors (%) of rotation axis $\hat{\mathbf{h}}$, rotation angle $\hat{\theta}$, and translation vector $\hat{\mathbf{t}}$ using synthetic points data with different motions.

n	$\theta(^\circ)$	k	Algo.	$e_{\mathbf{h}}(\%)$	$e_{\theta}(\%)$	$e_{\mathbf{t}}(\%)$
10	50		GA	3.61	-1.99	2.06
		1	k-GA	184.84	-44.78	9.26
		2	k-GA	5.13	-2.95	2.88
		4	k-GA	5.13	-2.95	2.88
		6	k-GA	5.13	-2.95	2.88
	70		GA	2.84	-1.29	2.18
		1	k-GA	140.81	-72.28	9.31
		2	k-GA	152.70	-64.91	15.55
		4	k-GA	3.75	-1.91	2.79
		6	k-GA	3.75	-1.91	2.79
12	10		GA	193.48	894.25	76.59
		2	k-GA	3.61	21.88	0.68
		4	k-GA	3.61	21.88	0.68
		6	k-GA	193.48	894.25	76.59
15	45		GA	0.72	-1.94	0.78
		1	k-GA	181.89	-72.93	12.22
		2	k-GA	1.50	-2.15	1.26
	60		GA	0.54	-1.58	0.81
		1	k-GA	86.74	-80.85	11.98
		2	k-GA	86.74	-80.85	11.98
		4	k-GA	1.06	-1.58	1.24

Table 4: The mean μ and standard deviation σ of relative calibration errors (%) of rotation axis $\hat{\mathbf{h}}$, rotation angle $\hat{\theta}$, and translation vector $\hat{\mathbf{t}}$, and calibration time in seconds using synthetic points data (n=90) with different motions and different levels of noise.

Noise	Meas.	Algo.	$e_{\mathbf{h}}(\%)$	$e_{\theta}(\%)$	$e_{\mathbf{t}}(\%)$	time (s)
σ_1	μ	GA	1.34	-1.17	0.19	12.70
		k-GA	1.34	-1.16	0.19	0.53
		GICP	6.31	2.47	0.85	0.94
	σ	GA	1.09	0.79	0.02	0.57
		k-GA	1.09	0.78	0.02	0.50
		GICP	4.82	3.37	0.25	0.54
σ_2	μ	GA	2.11	-1.39	0.28	12.76
		k-GA	2.18	-1.41	0.30	0.65
		GICP	8.20	1.20	0.97	1.35
	σ	GA	1.99	0.88	0.04	0.48
		k-GA	2.05	0.89	0.04	0.48
		GICP	9.47	5.43	0.36	0.90

Table 5: The mean μ and standard deviation σ of relative calibration errors (%) of rotation axis $\hat{\mathbf{h}}$, rotation angle $\hat{\theta}$, and translation vector $\hat{\mathbf{t}}$ and calibration time in seconds using synthetic points data with different numbers of points on the 3D free form shapes to be matched.

Meas.	Algo.	$e_{\mathbf{h}}(\%)$	$e_{\theta}(\%)$	$e_{\mathbf{t}}(\%)$	time (s)
μ	GA	1.25	-0.14	0.38	34.83
	k-GA	1.26	-0.14	0.38	0.86
	GICP	2.13	1.04	0.60	0.97
σ	GA	0.68	0.58	0.21	23.81
	k-GA	0.68	0.58	0.21	0.53
	GICP	1.53	4.60	0.45	0.44

Table 6: The number n of valid points and the average l_{μ} and standard deviation l_{δ} of the interpoint distances in millimetres in different range images.

Image	bird1	bird2	tubby1	tubby2
n	9234	10502	4354	4361
$l_{\mu}(mm)$	0.90	0.91	0.89	0.80
$l_{\delta}(mm)$	0.34	0.34	0.66	0.39

Table 7: The average e_μ and standard deviation e_σ of matching errors (mm), the number N of finally established reciprocal correspondences by the k-GA algorithm for the matching of 200 randomly selected points with uniform distribution from the bird1 and bunny1 images respectively.

Image	Registration	$e_\mu(mm)$	$e_\sigma(mm)$	N
bird1-2	before	1.42	1.01	48
	after	0.29	0.12	170
bunny1-2	before	0.90	0.65	29
	after	0.23	0.11	114

Table 8: The average e_μ and standard deviation e_σ of matching errors (mm), calibrated rotation angle $\hat{\theta}(^\circ)$ (20° is expected), the matching time in seconds, the number N of finally established reciprocal correspondences for different range images with small motions. The numbers in the parentheses are the numbers of sampled points for the GA algorithm.

Images	Alog.	$e_\mu(mm)$	$e_\sigma(mm)$	$\hat{\theta}(^\circ)$	$time(s)$	N
bird1-2	k-GA	0.28	0.12	19.82	176	7639
(112,128)	GA			15.40	19	
	GICP	0.31	0.12	19.22	46	7675
bird2-1	k-GA	0.28	0.12	19.80	207	7650
	GA			15.40	19	
	GICP	0.39	0.16	18.40	46	7432
tubby1-2	k-GA	0.25	0.14	19.67	69	2846
(55,57)	GA			3.52	4	
	GICP	0.38	0.21	16.60	11	2723
tubby2-1	k-GA	0.25	0.14	19.70	65	2857
	GA			3.52	4	
	GICP	0.26	0.14	19.45	9	2861

Table 9: The number n of valid points and the average l_μ and standard deviation l_δ of the interpoint distances in millimetres in different range images.

Image	bunny1	bunny2	dinosaur1	dinosaur2
n	6579	6460	8366	7107
$l_\mu(mm)$	0.65	0.68	1.63	1.63
$l_\delta(mm)$	0.24	0.32	1.40	1.93

Table 10: The average e_μ and standard deviation e_σ of matching errors (mm), calibrated rotation angle $\hat{\theta}(^\circ)$, the matching time in seconds, the number N of finally established reciprocal correspondences for different range images with large motions (Bunny and Dinosaur have an expected rotation angle of 40° and 36° around an unknown rotation axis respectively). The numbers in the parentheses are the numbers of sampled points for the GA algorithm.

Images	Alog.	$e_\mu(mm)$	$e_\sigma(mm)$	$\hat{\theta}(^\circ)$	time(s)	N
bunny1-2 (83,79)	k-GA	0.23	0.11	40.04	115	3748
	GA			36.59	9	
	GICP	0.25	0.12	39.07	22	3711
bunny2-1	k-GA	0.23	0.11	40.02	111	3750
	GA			36.59	9	
	GICP	0.35	0.23	35.88	24	3416
dinosaur1-2 (102,89)	k-GA	0.62	0.87	35.24	159	4311
	GA			34.32	12	
	GICP	1.12	0.86	22.47	29	2905
dinosaur2-1	k-GA	0.62	0.87	35.20	137	4297
	GA			34.32	12	
	GICP	1.32	0.96	25.30	52	2527

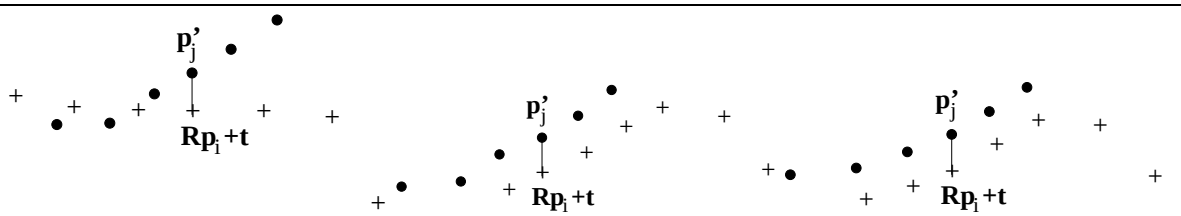

Figure 1: Physical interpretation for the various terms within the objective function of the GA algorithm. Left: first term is to minimize weighted matching errors; Middle: second term is to equalise weights; Right: third term is to maximise overlapping area.

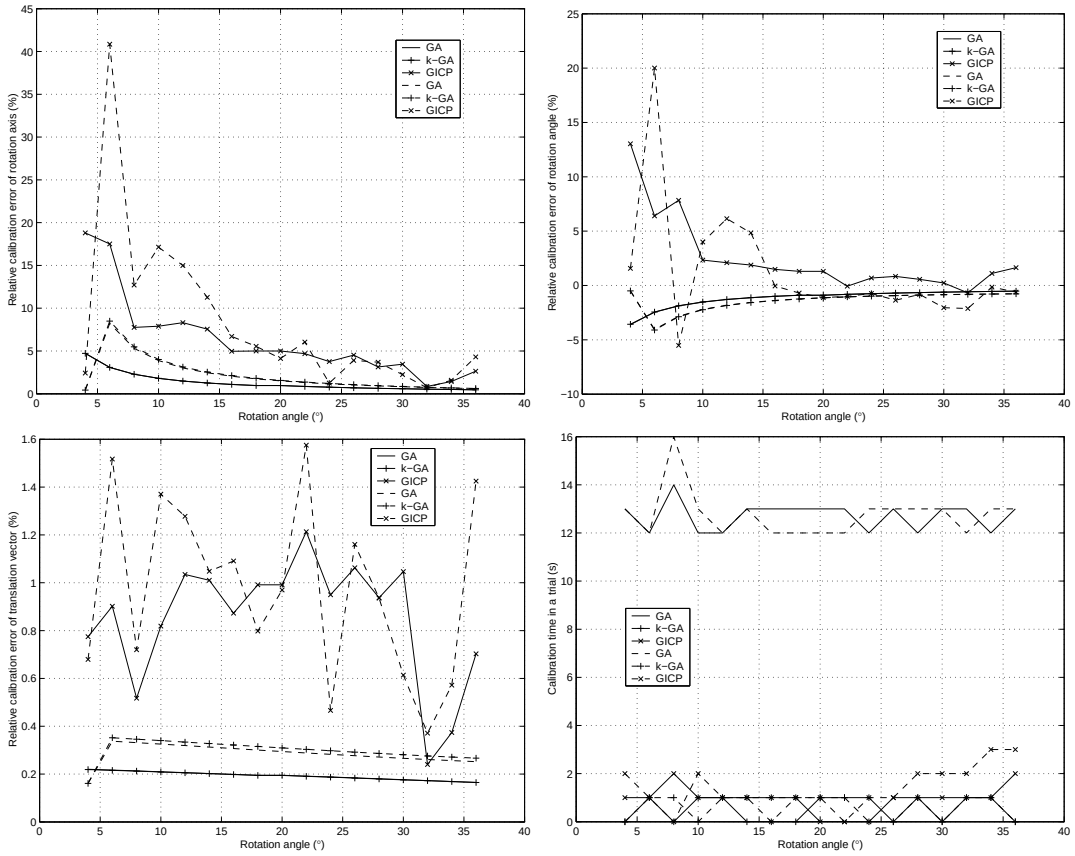

Figure 2: The relative calibration errors of the parameters of interest and calibration time as a function of the rotation angle. Top left: rotation axis; Top right: rotation angle; Bottom left: translation vector; Bottom right: the time in seconds for matching in a trial.

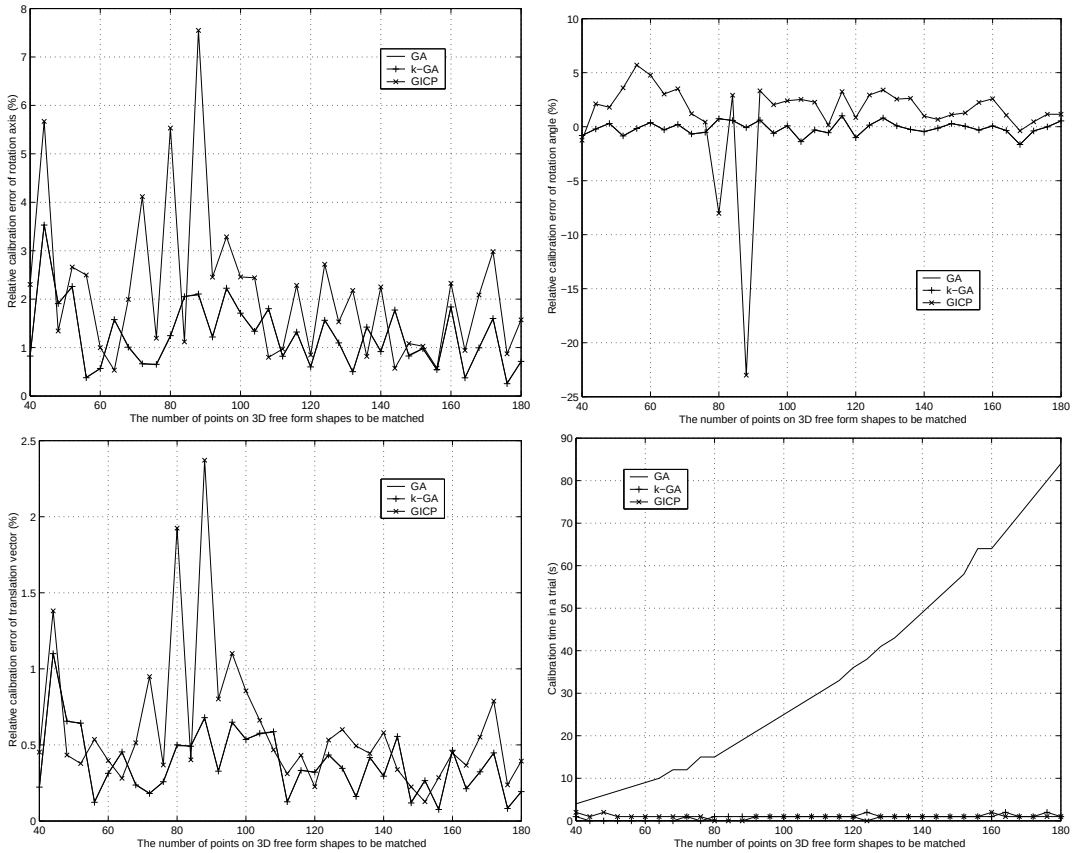

Figure 3: The relative calibration errors of the parameters of interest and calibration time as a function of the number of points on the 3D free form shapes to be matched. Top left: rotation axis; Top right: rotation angle; Bottom left: translation vector; Bottom right: the time in seconds for matching in a trial.

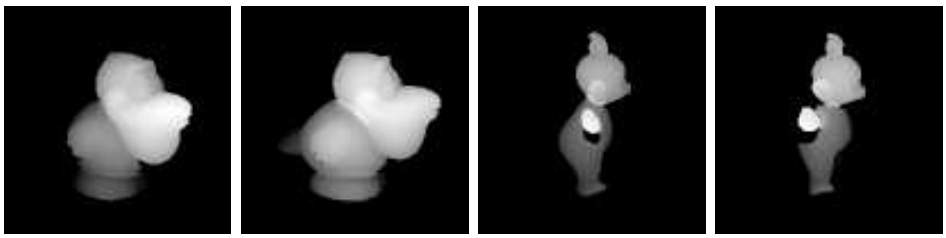

Figure 4: The real range images used. Left two: bird; Right two: tubby.

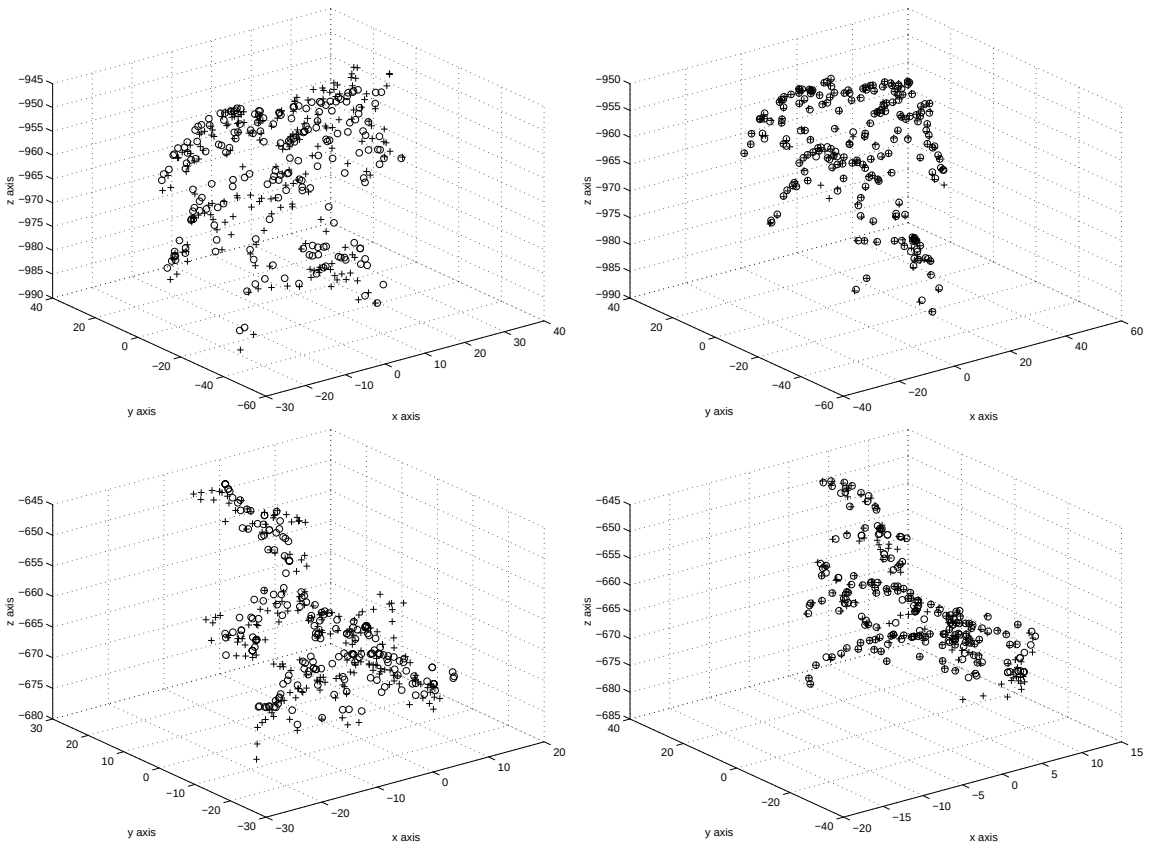

Figure 5: The results for the matching of 200 randomly selected points with uniform distribution from the bird1 and bunny1 images respectively by the k-GA algorithm. Left column: Before registration, Right column: after registration. Top row: bird1-2; bottom row: bunny1-2.

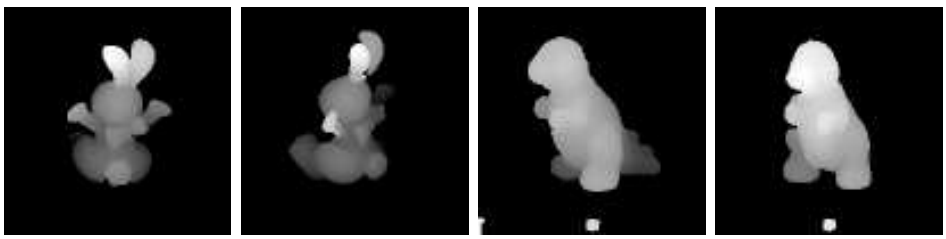

Figure 6: The real range images used. Left two: bunny; Right two: dinosaur.